

LAB 2

MỤC TIÊU

Kết thúc bài thực hành sinh viên có khả năng:

- ✓ Sử dụng RecyclerView
- ✓ Dùng Interface để xử lý các nút trong RecyclerView

Lưu ý: Sử dụng tiếp tục Project ở Lab 1

NỘI DUNG

BÀI 1: Hiển thị danh sách môn học bằng RecyclerView

Bước 1: Thay đổi layout item môn học như hình bên dưới:



Bước 2 : Tạo 1 adapter cho RecyclerView

```
public class ToDoAdapter extends RecyclerView.Adapter<ToDoAdapter.ViewHolder> {

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        return null;
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
    }

    @Override
    public int getItemCount() {
        return 0;
    }

    3 usages
    public class ViewHolder extends RecyclerView.ViewHolder {

        public ViewHolder(@NonNull View itemView) {
            super(itemView);
        }
    }
}
```

- Tạo constructor cho adapter với 3 tham số:

```
public class ToDoAdapter extends RecyclerView.Adapter<ToDoAdapter.ViewHolder> {  
    2 usages  
    private final Context context;  
    5 usages  
    private ArrayList<ToDo> list;  
    1 usage  
    private final ToDoDAO toDoDAO;  
  
    1 usage  
    public ToDoAdapter(Context context, ArrayList<ToDo> list, ToDoDAO toDoDAO) {  
        this.context = context;  
        this.list = list;  
        this.toDoDAO = toDoDAO;  
    }  
}
```

- Xử lý trong onCreateViewHolder

```
@NonNull  
@Override  
public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
    LayoutInflater inflater = ((Activity) context).getLayoutInflater();  
    View view = inflater.inflate(R.layout.item_todo, parent, attachToRoot: false);  
    return new ViewHolder(view);  
}
```

- Trong ViewHolder, khai báo và ánh xạ các widget có trong layout item_todo

```
public class ViewHolder extends RecyclerView.ViewHolder {  
    6 usages  
    TextView title, date;  
    3 usages  
    CheckBox cb_status;  
    1 usage  
    ImageView btn_xoa, btn_update;  
  
    1 usage  
    public ViewHolder(@NonNull View itemView) {  
        super(itemView);  
        title = itemView.findViewById(R.id.title);  
        date = itemView.findViewById(R.id.date);  
        cb_status = itemView.findViewById(R.id.cb_status);  
        btn_xoa = itemView.findViewById(R.id.btn_xoa);  
        btn_update = itemView.findViewById(R.id.btn_update);  
    }  
}
```

- Gán dữ liệu lên từng item trên RecyclerView

```
@Override  
public void onBindViewHolder(@NonNull ViewHolder holder, int position) {  
    //Set dữ liệu lên từng item trên recyclerview  
    holder.title.setText(list.get(position).getTitle());  
    holder.date.setText(list.get(position).getDate());  
  
    if (list.get(position).getStatus() == 1) {  
        holder.cb_status.setChecked(true);  
        holder.title.setPaintFlags(holder.title.getPaintFlags() | Paint.STRIKE_THRU_TEXT_FLAG);  
    } else {  
        holder.cb_status.setChecked(false);  
        holder.title.setPaintFlags(holder.title.getPaintFlags() & (~Paint.STRIKE_THRU_TEXT_FLAG));  
    }  
}
```

- Set số lượng item hiển thị lên RecyclerView

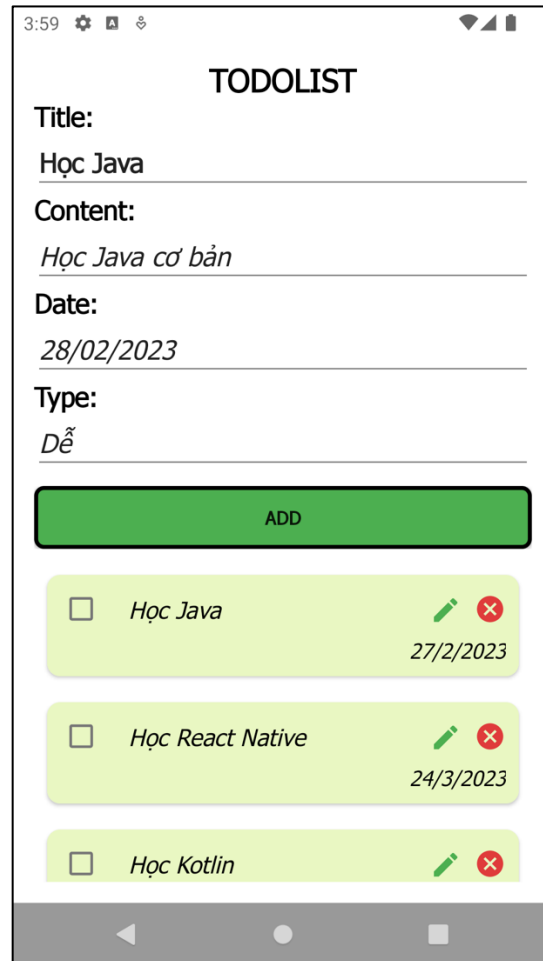
```
@Override
public int getItemCount() {
    //số lượng item
    return list.size();
}
```

Bước 3: Set adapter lên RecyclerView

```
LinearLayoutManager layoutManager = new LinearLayoutManager(context, this);
layoutManager.setOrientation(RecyclerView.VERTICAL);
recyclerViewListTodo.setLayoutManager(layoutManager);

ToDoDAO toDoDAO = new ToDoDAO(context, this);
ArrayList<ToDo> list = toDoDAO.getListTodo();
ToDoAdapter = new ToDoAdapter(context, this, list, toDoDAO);
recyclerViewListTodo.setAdapter(toDoAdapter);
```

Và đây thành quả



BÀI 2: Xử lý chức năng xóa một công việc có trong danh sách

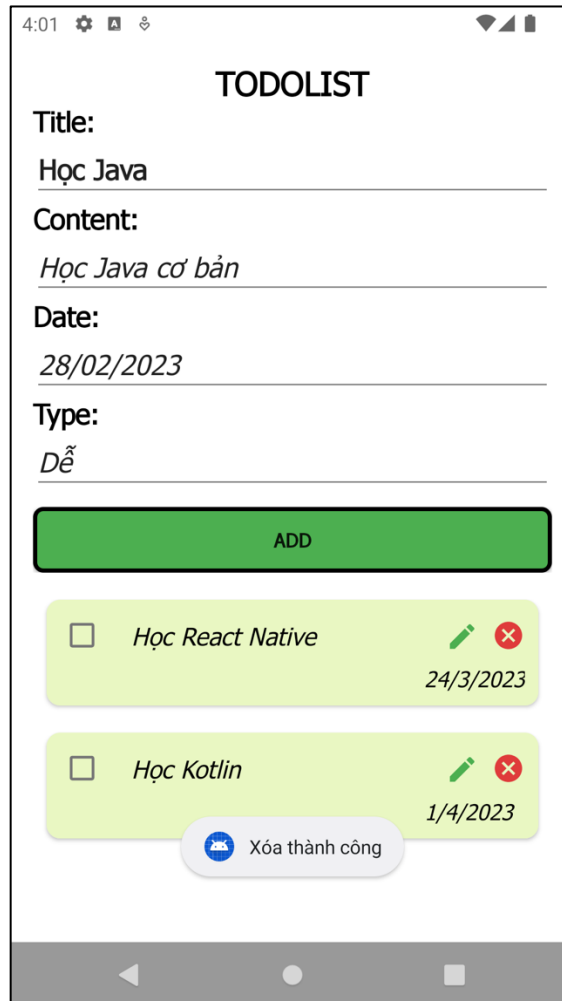
Bước 1: Trong **ToDoDAO**, viết thêm hàm xóa công việc trong danh sách

```
public boolean removeToDo(int id) {  
    SQLiteDatabase database = sqlite.getWritableDatabase();  
    int row = database.delete( table: "TODO", whereClause: "id = ?", new String[] {String.valueOf(id)});  
    return row != -1;  
}
```

Bước 2: Tại **onBindViewHolder** trong **ToDoAdapter**, xử lý sự kiện khi người dùng nhấn vào nút xóa công việc trên mỗi item

```
holder.btn_xoa.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        int id = list.get(holder.getAdapterPosition()).getId();  
        boolean check = todoDAO.removeToDo(id);  
        if (check) {  
            Toast.makeText(context, text: "Xóa thành công", Toast.LENGTH_SHORT).show();  
  
            //sau khi xóa thành công, load lại danh sách  
            list.clear();  
            list = todoDAO.getListTodo();  
            notifyItemRemoved(holder.getAdapterPosition());  
        } else {  
            Toast.makeText(context, text: "Xóa thất bại", Toast.LENGTH_SHORT).show();  
        }  
    }  
});
```

Kết quả khi ta nhấn nút xóa (dấu x), sẽ xóa đi một công việc trong danh sách



BÀI 3: Xây dựng chức năng update trạng thái công việc

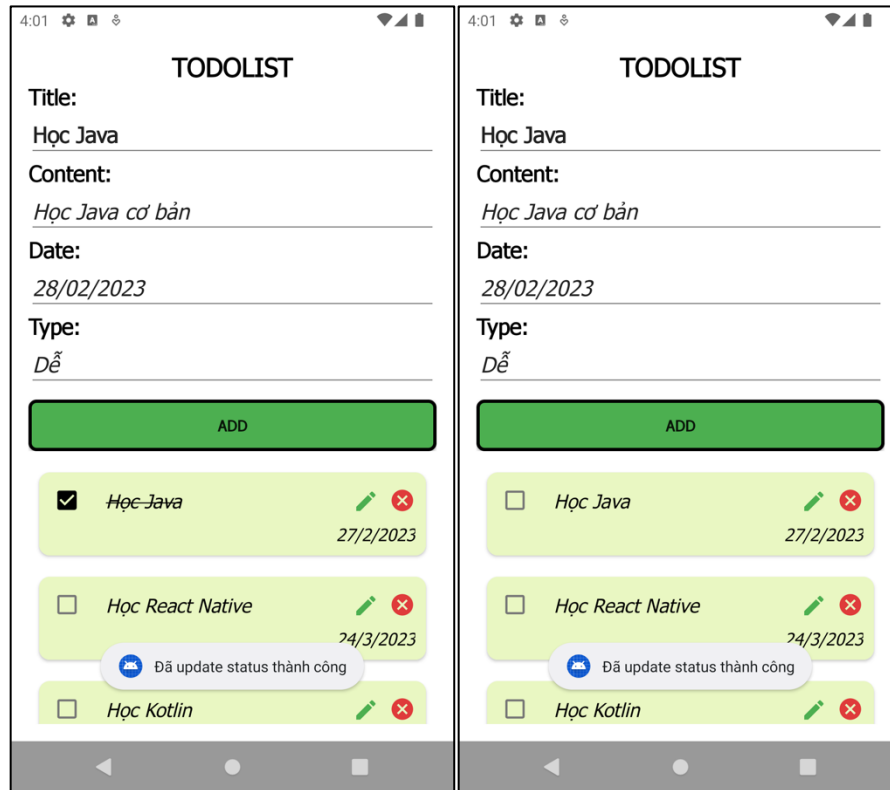
Bước 1: Trong **ToDoDAO**, viết thêm hàm cập nhật tình trạng công việc

```
public boolean updateStatusToDo(Integer id, boolean check) {
    SQLiteDatabase database = sqlite.getWritableDatabase();
    //Chúng ta sẽ dùng biến check truyền vào để dựa vào đó lấy kết quả set cho status của todo
    //SQLite không có kiểu dữ liệu boolean nên chúng ta sẽ tự quy định:
    // 1 là true (đã hoàn thành công việc)
    // 0 là false (chưa hoàn thành công việc)
    int statusValue = check ? 1 : 0;
    //Ta sẽ sử dụng contentValues để cập nhật
    ContentValues values = new ContentValues();
    values.put("STATUS", statusValue);
    long row = database.update( table: "TODO", values, whereClause: "id = ?", new String[]{String.valueOf(id)});
    return row != -1;
}
```

Bước 2: Tại **onBindViewHolder** trong **ToDoAdapter**, xử lý sự kiện khi người dùng check vào checkbox trên mỗi item công việc

```
holder.cb_status.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton compoundButton, boolean b) {
        int id = list.get(holder.getAdapterPosition()).getId();
        boolean check_result = todoDAO.updateStatusToDo(id, holder.cb_status.isChecked());
        if (check_result) {
            Toast.makeText(context, text: "Đã update status thành công", Toast.LENGTH_SHORT).show();
            list.clear();
            list = todoDAO.getListTodo();
            notifyDataSetChanged();
        } else {
            Toast.makeText(context, text: "Update status Thất bại", Toast.LENGTH_SHORT).show();
        }
    }
});
```

Kết quả ta được



BÀI 4: GV CHO THÊM

*** YÊU CẦU NỘP BÀI:

Sv nén file bao gồm các yêu cầu đã thực hiện trên, nộp lms đúng thời gian quy định của giảng viên. Không nộp bài coi như không có điểm.

--- Hết ---